AMENDMENTS TO THE CLAIMS

The listing of claims will replace all prior versions, and listings of claims in the application:

LISTING OF CLAIMS

1 (Original) A method for arithmetic expression optimization, comprising:

receiving a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand;

converting said first instruction to a second instruction optimized for a second processor having a second base when said at least one operand does not carry potential overflow beyond said second base or when said operator is insensitive to overflow, said second base smaller than said first base; and

converting to a wider base a third instruction that is the source of potential overflow associated with said at least one operand when said at least one operand carries the potential for overflow beyond said second base and when said operator is sensitive to overflow, said third instruction having been previously optimized, said wider base larger than said second base and smaller or equal to said first base.

2-35. (Cancelled)

36. (New) A method for arithmetic expression optimization, comprising:

receiving a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand; and

converting said first instruction to a second instruction optimized for a second processor having a second base when said at least one operand does not carry

potential overflow beyond said second base or when said operator is insensitive to overflow, said second base smaller than said first base.

37. (New) The method of claim 36, further comprising, after said converting said first instruction, returning to receiving said first instruction until all instructions defined for said first processor are converted.

38. (New) The method of claim 36, further comprising rejecting an expression that cannot be optimized to a smaller base on said second processor.

39. (New) The method of claim 36 wherein said first instruction is arithmetic.

40. (New) The method of claim 36 wherein said first instruction comprises a non-arithmetic, type-sensitive instruction.

41. (New) The method of claim 36, further comprising linking each instruction to successor instructions in all control paths.

42. (New) The method of claim 36 wherein said converting said first instruction further comprises:
linking each result of an instruction to all instructions that consume said result;
if said converting includes creating a value, linking said value to the instruction that produced said value; and
if said value carries a potential for overflow, linking said value to the instruction that originally caused said overflow.

43. (New) The method of claim 36 wherein
said first processor comprises a Java™ Virtual Machine; and
said second processor comprises a Java Card™ Virtual Machine.

44. (New) The method of claim 36 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size equal to the size of said at least one data type.

45. (New) The method of claim 36 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size greater than the size of said at least one data type.

46. (New) A method for arithmetic expression optimization, comprising:

receiving a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand; and

converting to a wider base a third instruction that is the source of potential overflow associated with said at least one operand when said at least one operand carries the potential for overflow beyond a second base of a second processor and when said operator is sensitive to overflow, said third instruction having been previously optimized, said second base smaller than said first base, said wider base larger than said second base and smaller or equal to said first base.

47. (New) The method of claim 46 wherein said converting to a wider base further comprises discarding previous conversion results of said third instruction before said converting to a wider base.

48. (New) The method of claim 46, further comprising rejecting an expression that cannot be optimized to a smaller base on said second processor.

49. (New) The method of claim 46 wherein said converting to a wider base further comprises rejecting said first instruction when said wider base is not supported by said second processor.

50. (New) The method of claim 46 wherein said first instruction is arithmetic.

51. (New) The method of claim 46 wherein said first instruction comprises a non-arithmetic, type-sensitive instruction.

52. (New) The method of claim 46, further comprising linking each instruction to successor instructions in all control paths.

53. (New) The method of claim 46 wherein
said first processor comprises a Java™ Virtual Machine; and
said second processor comprises a Java Card™ Virtual Machine.

54. (New) The method of claim 46 wherein
said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and
said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size equal to the size of said at least one data type.

55. (New) The method of claim 46 wherein
said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size greater than the size of said at least one data type.

56. (New) A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a method for arithmetic expression optimization, the method comprising:
receiving a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand; and
converting said first instruction to a second instruction optimized for a second processor having a second base when said at least one operand does not carry potential overflow beyond said second base or when said operator is insensitive to overflow, said second base smaller than said first base.

57. (New) The program storage device of claim 56, said method further comprising, after said converting said first instruction, returning to receiving said first instruction until all instructions defined for said first processor are converted.

58. (New) The program storage device of claim 56, said method further comprising rejecting an expression that cannot be optimized to a smaller base on said second processor.

59. (New) The program storage device of claim 56 wherein said first instruction is arithmetic.

60. (New) The program storage device of claim 56 wherein said first instruction comprises a non-arithmetic, type-sensitive instruction.

61. (New) The program storage device of claim 56, said method further comprising linking each instruction to successor instructions in all control paths.

62. (New) The program storage device of claim 56 wherein said converting said first instruction further comprises:

linking each result of an instruction to all instructions that consume said result;

if said converting includes creating a value, linking said value to the instruction that produced said value; and

if said value carries a potential for overflow, linking said value to the instruction that originally caused said overflow.

63. (New) The program storage device of claim 56 wherein

said first processor comprises a Java™ Virtual Machine; and

said second processor comprises a Java Card™ Virtual Machine.

64. (New) The program storage device of claim 56 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size equal to the size of said at least one data type.

65. (New) The program storage device of claim 56 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size greater than the size of said at least one data type.

66. (New) A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a method for arithmetic expression optimization, the method comprising:

receiving a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand; and

converting to a wider base a third instruction that is the source of potential overflow associated with said at least one operand when said at least one operand carries the potential for overflow beyond a second base of a second processor and when said operator is sensitive to overflow, said third instruction having been previously optimized, said second base smaller than said first base, said wider base larger than said second base and smaller or equal to said first base.

67. (New) The program storage device of claim 66 wherein said converting to a wider base further comprises discarding previous conversion results of said third instruction before said converting to a wider base.

68. (New) The program storage device of claim 66, said method further comprising rejecting an expression that cannot be optimized to a smaller base on said second processor.

69. (New) The program storage device of claim 66 wherein said converting to a wider base further comprises rejecting said first instruction when said wider base is not supported by said second processor.

70. (New) The program storage device of claim 66 wherein said first instruction is arithmetic.

71. (New) The program storage device of claim 66 wherein said first instruction comprises a non-arithmetic, type-sensitive instruction.

72. (New) The program storage device of claim 66, said method further comprising linking each instruction to successor instructions in all control paths.

73. (New) The program storage device of claim 66 wherein

said first processor comprises a Java™ Virtual Machine; and

said second processor comprises a Java Card™ Virtual Machine.

74. (New) The program storage device of claim 66 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size equal to the size of said at least one data type.

75. (New) The program storage device of claim 66 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size greater than the size of said at least one data type.

76. (New) An apparatus for arithmetic expression optimization, comprising:

means for receiving a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand; and

means for converting said first instruction to a second instruction optimized for a second processor having a second base when said at least one operand does not carry potential overflow beyond said second base or when said operator is insensitive to overflow, said second base smaller than said first base.

77. (New) The apparatus of claim 76, further comprising means for, after said converting said first instruction, returning to receiving said first instruction until all instructions defined for said first processor are converted.

78. (New) The apparatus of claim 76, further comprising means for rejecting an expression that cannot be optimized to a smaller base on said second processor.

79. (New) The apparatus of claim 76 wherein said first instruction is arithmetic.

80. (New) The apparatus of claim 76 wherein said first instruction comprises a non-arithmetic, type-sensitive instruction.

81. (New) The apparatus of claim 76, further comprising means for linking each instruction to successor instructions in all control paths.

82. (New) The apparatus of claim 76 wherein said means for converting said first instruction further comprises:

means for linking each result of an instruction to all instructions that consume said result;

means for, if said converting includes creating a value, linking said value to the instruction that produced said value; and

means for, if said value carries a potential for overflow, linking said value to the instruction that originally caused said overflow.

83. (New) The apparatus of claim 76 wherein

said first processor comprises a Java™ Virtual Machine; and

said second processor comprises a Java Card™ Virtual Machine.

84. (New) The apparatus of claim 76 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size equal to the size of said at least one data type.

85. (New) The apparatus of claim 76 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size greater than the size of said at least one data type.

86. (New) An apparatus for arithmetic expression optimization, comprising:

means for receiving a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand; and

means for converting to a wider base a third instruction that is the source of potential overflow associated with said at least one operand when said at least one operand carries the potential for overflow beyond a second base of a second processor and when said operator is sensitive to overflow, said third instruction having been previously optimized, said second base smaller than said first base, said wider base larger than said second base and smaller or equal to said first base.

87. (New) The apparatus of claim 86 wherein said means for converting to a wider base further comprises means for discarding previous conversion results of said third instruction before said converting to a wider base.

88. (New) The apparatus of claim 86, further comprising means for rejecting an expression that cannot be optimized to a smaller base on said second processor.

89. (New) The apparatus of claim 86 wherein said means for converting to a wider base further comprises means for rejecting said first instruction when said wider base is not supported by said second processor.

90. (New) The apparatus of claim 86 wherein said first instruction is arithmetic.

91. (New) The apparatus of claim 86 wherein said first instruction comprises a non-arithmetic, type-sensitive instruction.

92. (New) The apparatus of claim 86, further comprising means for linking each instruction to successor instructions in all control paths.

93. (New) The apparatus of claim 86 wherein
said first processor comprises a Java™ Virtual Machine; and
said second processor comprises a Java Card™ Virtual Machine.

94. (New) The apparatus of claim 86 wherein
said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and
said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size equal to the size of said at least one data type.

95. (New) The apparatus of claim 86 wherein
said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and
said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size greater than the size of said at least one data type.

96. (New) An apparatus for arithmetic expression optimization, comprising:
at least one memory having program instructions; and
at least one processor configured to use the program instructions to:

receive a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand; and

convert said first instruction to a second instruction optimized for a second processor having a second base when said at least one operand does not carry potential overflow beyond said second base or when said operator is insensitive to overflow, said second base smaller than said first base.

97. (New) The apparatus of claim 96 wherein said apparatus is further configured to, after said converting said first instruction, return to receiving said first instruction until all instructions defined for said first processor are converted.

98. (New) The apparatus of claim 96, wherein said apparatus is further configured to reject an expression that cannot be optimized to a smaller base on said second processor.

99. (New) The apparatus of claim 96 wherein said first instruction is arithmetic.

100. (New) The apparatus of claim 96 wherein said first instruction comprises a non-arithmetic, type-sensitive instruction.

101. (New) The apparatus of claim 96 wherein said apparatus is further configured to link each instruction to successor instructions in all control paths.

102. (New) The apparatus of claim 96 wherein said apparatus is further configured to convert said first instruction by:

linking each result of an instruction to all instructions that consume said result;

if said converting includes creating a value, linking said value to the instruction that produced said value; and

if said value carries a potential for overflow, linking said value to the instruction that originally caused said overflow.

103. (New) The apparatus of claim 96 wherein

said first processor comprises a Java™ Virtual Machine; and

said second processor comprises a Java Card™ Virtual Machine.

104. (New) The apparatus of claim 96 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size equal to the size of said at least one data type.

105. New) The apparatus of claim 96 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size greater than the size of said at least one data type.

106. (New) An apparatus for arithmetic expression optimization, comprising:

at least one memory having program instructions; and

at least one processor configured to use the program instructions to:

receive a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand; and

convert to a wider base a third instruction that is the source of potential overflow associated with said at least one operand when said at least one operand carries the potential for overflow beyond a second base of a second processor and when said operator is sensitive to overflow, said third instruction having been previously optimized, said second base smaller than said first base, said wider base larger than said second base and smaller or equal to said first base.

107. (New) The apparatus of claim 106 wherein said apparatus is further configured to convert to a wider base comprises discarding previous conversion results of said third instruction before said converting to a wider base.

108. (New) The apparatus of claim 106 wherein said apparatus is further configured to reject an expression that cannot be optimized to a smaller base on said second processor.

109. (New) The apparatus of claim 106 wherein said converting to a wider base further comprises rejecting said first instruction when said wider base is not supported by said second processor.

110. (New) The apparatus of claim 106 wherein said first instruction is arithmetic.

111. (New) The apparatus of claim 106 wherein said first instruction comprises a non-arithmetic, type-sensitive instruction.

112. (New) The apparatus of claim 106 wherein said apparatus is further configured to link further comprising linking each instruction to successor instructions in all control paths.

113. (New) The apparatus of claim 106 wherein
said first processor comprises a Java™ Virtual Machine; and
said second processor comprises a Java Card™ Virtual Machine.

114. (New) The apparatus of claim 106 wherein
said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size equal to the size of said at least one data type.

115. (New) The apparatus of claim 106 wherein

said first base is used by said first processor for performing arithmetic operations on at least one data type, said at least one data type having a size less than the size of said first base; and

said second base is used by said second processor for performing arithmetic operations on said at least one data type, said second base having a size greater than the size of said at least one data type.

116. (New) A smart card having a microcontroller embedded therein, the smart card comprising a virtual machine being executed by a microcontroller, the virtual machine executing a software application comprising of a plurality of previously optimized instructions, the instructions optimized by a method comprising:

receiving a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand; and

converting said first instruction to a second instruction optimized for a second processor having a second base when said at least one operand does not carry potential overflow beyond said second base or when said operator is insensitive to overflow, said second base smaller than said first base;

the virtual machine comprising means for receiving optimized instructions, the optimized

instructions being previously optimized for execution on a resource-constrained deviceand means for executing said instructions.

117. (New) A smart card having a microcontroller embedded therein, the smart card comprising a virtual machine being executed by a microcontroller, the virtual machine executing a software application comprising of a plurality of previously optimized instructions, the instructions optimized by a method comprising:

receiving a first instruction defined for a first processor having a first base, said instruction including an operator and at least one operand; and

converting to a wider base a third instruction that is the source of potential overflow associated with said at least one operand when said at least one operand carries the potential for overflow beyond a second base of a second processor and when said operator is sensitive to overflow, said third instruction having been previously optimized, said second base smaller than said first base, said wider base larger than said second base and smaller or equal to said first base.

the virtual machine comprising means for receiving optimized instructions, the optimized

instructions being previously optimized for execution on a resource-constrained device

and means for executing said instructions.